Scientific Excellence • Resource Protection & Conservation • Benefits for Canadians
Excellence scientifique • Protection et conservation des ressources • Bénéfices aux Canadiens

A Fortran Program for Fitting Selectivity Curves to Trouser Trawl Data

R. B. Millar and N. G. Cadigan

Science Branch
Department of Fisheries and Oceans
P. O. Box 5667
St. John's, Newfoundland A1C 5X1

February 1991

Canadian Technical Report of Fisheries and Aquatic Sciences No. 1783

Fisheries and Oceans Pêches et Océans

Canada

Canadian Technical Report of Fisheries and Aquatic Sciences

Technical reports contain scientific and technical information that contributes to existing knowledge but which is not normally appropriate for primary literature. Technical reports are directed primarily toward a worldwide audience and have an international distribution. No restriction is placed on subject matter and the series reflects the broad interests and policies of the Department of Fisheries and Oceans, namely, fisheries and aquatic sciences.

Technical reports may be cited as full publications. The correct citation appears above the abstract of each report. Each report is abstracted in *Aquatic Sciences and Fisheries Abstracts* and indexed in the Department's annual index to scientific and technical publications.

Numbers 1–456 in this series were issued as Technical Reports of the Fisheries Research Board of Canada. Numbers 457–714 were issued as Department of the Environment, Fisheries and Marine Service, Research and Development Directorate Technical Reports. Numbers 715–924 were issued as Department of Fisheries and the Environment, Fisheries and Marine Service Technical Reports. The current series name was changed with report number 925.

Technical reports are produced regionally but are numbered nationally. Requests for individual reports will be filled by the issuing establishment listed on the front cover and title page. Out-of-stock reports will be supplied for a fee by commercial agents.

Rapport technique canadien des sciences halieutiques et aquatiques

Les rapports techniques contiennent des renseignements scientifiques et techniques qui constituent une contribution aux connaissances actuelles, mais qui ne sont pas normalement appropriés pour la publication dans un journal scientifique. Les rapports techniques sont destinés essentiellement à un public international et ils sont distribués à cet échelon. Il n'y a aucune restriction quant au sujet; de fait, la série reflète la vaste gamme des intérêts et des politiques du ministère des Pêches et des Océans, c'est-à-dire les sciences halieutiques et aquatiques.

Les rapports techniques peuvent être cités comme des publications complètes. Le titre exact paraît au-dessus du résumé de chaque rapport. Les rapports techniques sont résumés dans la revue *Résumés des sciences aquatiques et halieutiques*, et ils sont classés dans l'index annual des publications scientifiques et techniques du Ministère.

Les numéros 1 à 456 de cette série ont été publiés à titre de rapports techniques de l'Office des recherches sur les pêcheries du Canada. Les numéros 457 à 714 sont parus à titre de rapports techniques de la Direction générale de la recherche et du développement, Service des pêches et de la mer, ministère de l'Environnement. Les numéros 715 à 924 ont été publiés à titre de rapports techniques du Service des pêches et de la mer, ministère des Pêches et de l'Environnement. Le nom actuel de la série a été établi lors de la parution du numéro 925.

Les rapports techniques sont produits à l'échelon régional, mais numérotés à l'échelon national. Les demandes de rapports seront satisfaites par l'établissement auteur dont le nom figure sur la couverture et la page du titre. Les rapports épuisés seront fournis contre rétribution par des agents commerciaux.

91-04664/

Canadian Technical Report of
Fisheries and Aquatic Sciences 1783

February 1991

A FORTRAN PROGRAM FOR FITTING SELECTIVITY CURVES TO TROUSER TRAWL DATA

by

R. B. Millar and N. G. Cadigan

Science Branch
Department of Fisheries and Oceans
P.O Box 5667
St John's, Newfoundland A1C 5X1
Canada

ABSTRACT

Millar, R. B., and N. G. Cadigan. 1991. A FORTRAN program for fitting selectivity curves to trouser trawl data. Can. Tech. Rep. Fish. Aquat. Sci. 1783: iii + 20 p.

This technical report describes a FORTRAN program (TTRAWL.FOR) to fit a logistic selectivity curve to trouser trawl data using the method developed by Millar and Walsh (1990). This program is needed because the Millar and Walsh method has previously been fully implemented only in the statistical software package SPLUS, which may be unavailable or unfamiliar to many researchers. Other packages (e.g. SAS, SPSS, GLIM) do not permit easy implementation of the model, if at all. Output from TTRAWL.FOR includes estimates of the selectivity curve parameters, the lengths corresponding to 25, 50 and 75 percent retention, and their asymptotic standard errors. The program can estimate the relative fishing efficiency of the two codends (small mesh and large mesh) and provides goodness of fit statistics. The estimated retention probabilities are output to a separate file to facilitate plotting of the fitted selectivity curve.

RÉSUMÉ

Millar, R. B., and N. G. Cadigan. 1991. A FORTRAN program for fitting selectivity curves to trouser trawl data. Can. Tech. Rep. Fish. Aquat. Sci. 1783: iii + 20 p.

Ce rapport technique décrit un programme en FORTRAN (TTRAWL.FOR) servant à ajuster une courbe de sélectivité logistique aux données sur les chaluts pantalons au moyen de la méthode de Millar et Walsh (1990). Ce programme est nécessaire parce que la méthode de Millar et Walsh n'a été utilisée à fond jusqu'ioi que dans le progiciel SPLUS, lequel peut s'avérer non disponible pour de nombreux chercheurs ou inconnu d'eux. Avec d'autres progiciels (par exemple, SAS, SPSS et GLIM), le modèle ne peut étre utilisé que difficilement dans les meilleurs des cas. Le programme TTRAWL.FOR calcule entre autres les paramètres de la courbe de sélectivité, les longueurs correspondant à des taux de rétention de 25, 50 et 75 % et leurs écarts types asymptotiques. Il peut évaluer également les rendements de pêche relatifs des deux culs-de-chalut (petites mailles et grandes mailles) et la qualité de l'ajustement. Les taux de rétention calculés sont versés dans un fichier distinct pour faciliter le traçage de la courbe de sélectivé

©Minister of Supply and Services Canada 1990
Cat. No. Fs 97-6/1783E ISSN 0706-6457

Correct citation for this publication:

Millar, R. B., and N. G. Cadigan. 1991. A FORTRAN program for fitting selectivity curves to trouser trawl data. Can. Tech. Rep. Fish. Aquat. Sci. 1783: iii + 20 p.

INTRODUCTION

Program TTRAWL.FOR is appropriate for the analysis of data gathered from trouser trawl, twin trawl, or alternate haul surveys, in which a small mesh codend (assumed to be non-selective) is fished in conjunction with a larger mesh codend. Here the small mesh codend will be referred to as the control codend since its selectivity is assumed to be known and equal to 100% for all length-classes of fish used in the analysis. The large mesh codend will be referred to as the experimental codend. The analysis performed by TTRAWL.FOR produces an estimated selectivity curve for the experimental codend and an estimate of its fishing efficiency with respect to the control codend.

Data obtained from the above types of survey can not be properly analysed by the standard statistical software packages because they do not conform to the assumptions required by conventional statistical methodology for analyzing count data. The difficulty is that the number of fish entering the experimental codend (large mesh) is not observed. Previous analyses have typically assumed that the number of fish in the control codend (small mesh) is either the same as, or proportional to (if fishing efficiency of the codends is not equal), the number that entered the experimental codend (Pope et al. 1975). Of course, this assumption does not consider random variation in the catch of each codend. This can lead to very severe errors in fitting the selectivity curve (Millar and Walsh 1990). The method used by Millar and Walsh models the data correctly and therefore permits a full statistical analysis of the data, including calculation of standard errors and goodness of fit statistics. In addition, the assumption that the experimental and control codends are fishing equally can be formally tested.

Note that TTRAWL.FOR is not appropriate for data collected from covered codend experiments. This data can be analyzed by conventional methods because the number of fish entering the experimental codend is observed (the total of fish in the codend and cover).

PROGRAM SUMMARY

ESTIMATED SELECTIVITY CURVE

TTRAWL.FOR uses conditional maximum likelihood to fit a logistic selectivity curve,

$$r(l) = \frac{\exp(a + bl)}{1 + \exp(a + bl)}, \tag{1}$$

where $r(l)$ is the retention probability for a length l fish that enters the experimental codend. Details of the model and fitting procedure are given in Millar and Walsh (1990). The user can request TTRAWL.FOR to assume equal fishing efficiency of the two codends ($p = 0.5$, where p is the probability that a fish encountering the gear enters the experimental codend) or to estimate p. In TTRAWL.FOR output these are referred to as the two parameter and three parameter models respectively.

Two output files are produced. File `params.out` (Appendix 3) contains the estimates of the parameters a and b (and p if estimated) and their asymptotic variances and covariances. It also gives estimates of the retention lengths l_{25}, l_{50}, l_{75}, and their asymptotic variances, and the deviance (likelihood ratio statistic) for the fitted selectivity curve. The deviance can be used as a goodness of fit statistic, and under the hypothesis of correct model specification its distribution is approximately chi-square. However, Mc Cullagh and Nelder (1989, pg 119) caution that the approximation to chi-square can be inaccurate and that the deviance should be used only as a rough guide to model suitability. The degrees of freedom stated in `params.out` is the number of length-classes with non-zero catch (in either codend) minus the number of parameters. The hypothesis that the split parameter p equals 0.5 can be tested using the reduction in deviance between the two parameter and three parameter models. This is approximately distributed as chi-squared with one degree of freedom. File `rprobs.out` (Appendix 3) contains the estimated retention probabilities calculated from equation (1) above, and the deviance residual (Mc Cullagh and Nelder 1989, pg 39) for each length-class. Inspection of these residuals provides a further check of model suitability.

Note that for the example in Appendix 3 p is estimated to be 0.585 and that it is significantly different from 0.5 (the reduction in deviance between the two and three parameter models is 9.15 on one degree of freedom). For this reason the selectivity curve from the three parameter model should be used for this data set (Appendix 1).

ASSUMPTIONS

Program TTRAWL.FOR assumes that fish encountering the gear behave independently of each other. This may be questionable, especially for schooling fish. When not behaving independently the consequence is "over-dispersion" (McCullagh and Nelder 1989, pg 124). The parameter and retention length estimates remain valid but the estimates of their asymptotic variances tend to be too small.

The user can specify whether the two codends (experimental and control) are assumed to have equal fishing efficiency or whether the split of fish into the experimental codend is to be estimated. It is implicitly assumed that the split probability is independent of fish length.

If the trouser trawl data is an estimate of total catch derived from a sample of the catch then the parameter and retention length estimates remain valid but their asymptotic variances will not. However, reasonable estimates of the variances should be obtainable via a suitable adjustment that will depend upon the proportion of the total catch that was sampled. An implicit way of implementing this adjustment is to scale the catch data prior to analysis. For example, if x and y are the respective proportions of the control codend and experimental codend catches that were sampled then we recommend multiplying the estimated total catches in both codends by the lesser of x and y. (This may result in non-integer catch figures, which are permitted by TTRAWL.FOR.)

IMPLEMENTATION

TTRAWL.FOR was developed on a SUN Microsystem's UNIX workstation using a SUN Fortran compiler and was also run successfully, without modification, on a VAX 6310 running VMS version 5.3-1 and on an IBM PC clone using Microsoft Fortran. The maximum likelihood estimates are obtained using the Newton-Raphson method with analytic derivatives. Dennis and Schnabel (1983) show that this technique is locally convergent only. The algorithm requires initial estimates of the model parameters that are in a reasonably close neighborhood of the maximizing values. For this reason the user is required to supply initial estimates of the retention lengths l_{25} and l_{50} (must be positive), from which the initial values for a and b are computed. The data should suggest reasonable estimates for l_{25} and l_{50}; however, if the algorithm does not converge then TTRAWL.FOR should be rerun with different initial values. In addition, the user is required to input the convergence tolerance and the maximum number of iterations the algorithm is allowed.

Generally a convergence tolerance of 0.0000001 gives good results. The maximum number of iterations will prevent a diverging solution from running too long.

The program has been tested on a variety of data sets (cod, herring, haddock, American plaice, etc.) using the above convergence tolerance and a maximum number of iterations of 500. For most of the data sets the algorithm is quite robust in that the same solution is reached for a variety of starting values in less than 100 iterations (sometimes much less). Only one data set required use of starting values other than those initially guessed. On the VAX mainframe and the UNIX workstation TTRAWL.FOR generally completed execution in about one second. Run times ranged from 3 to 25 seconds on the PC.

INSTRUCTIONS FOR RUNNING TTRAWL.FOR

TTRAWL.FOR runs quickly and so has been implemented to run interactively. The program makes some checks of the interactive user supplied input. An input file containing the trouser trawl data is required (Appendix 1). This file contains three columns in free format: length-class, experimental codend catch numbers, and control codend catch numbers respectively. The maximum number of length-classes permitted is 100. An example of the user required interactive input is given in Appendix 2.

REFERENCES

Dennis, J. E., Jr. and R. B. Schnabel. 1983. Numerical Methods for Unconstrained Optimization and Nonlinear Equations. Prentice-Hall Inc. 378 p.

Mc Cullagh, P. and J. A. Nelder. 1989. Generalized Linear Models, 2nd edition. Chapman and Hall. 511 p.

Millar, R. B., and S. J. Walsh. 1990. Analysis of trawl selectivity studies with an application to trouser trawls. ICES C.M. 1990/B: 14: 14 p.

Pope, J. A., A. R. Margetts, J. M. Hamley and E. F. Akyuz. 1975. Manual of methods for fish stock assessment, Part III. Selectivity of fishing gear. FAO Fish. Tech. Pap. 41: 65 p.

APPENDIX 1

Sample input file (P88155.DATA). Columns 2 and 3 give the numbers in the large (experimental) and small (control) mesh codends respectively, for each lengthclass in column 1. Data is total count of American plaice (*Hippoglossoides platessoides*) summed over thirteen hauls in a 1988 trouser trawl survey. The experimental and control meshes were 155mm diamond and 38mm diamond respectively.

8	0	0
10	0	1
12	0	0
14	0	1
16	0	3
18	0	4
20	0	6
22	0	5
24	1	11
26	0	17
28	1	26
30	5	39
32	18	57
34	20	69
36	39	74
38	56	69
40	58	64
42	52	62
44	60	53
46	55	48
48	39	41
50	28	23
52	22	15
54	15	5
56	11	10
58	10	5
60	11	4
62	10	3
64	8	3
66	2	2
68	0	1
70	3	1
72	0	0
74	1	0

APPENDIX 2

An example run of TTRAWL.FOR using the trouser trawl data from Appendix 1. To distinguish user input from program output the user input has been preceded by the prompt ? in this example.

```
            Input name of data file

? p88155.data

          Input initial guesses of L.25 and L.50

? 30 35

    Input convergence tolerance and maximum number of iterations

? 0.0000001 100

                  Choose which model(s) to fit

                  Enter 1,2 or 3

                  1 : 50:50 split

                  2 : Estimated split

                  3 : Both 1 and 2

? 3

 Program finished:
  - estimated statistics are in file  params.out
  - estimated selectivity curve(s) and deviance residuals
    are in file rprobs.out
```

APPENDIX 3

Listing of the two files produced by TTRAWL.FOR for the run generated from Appendix 2. File params.out contains the fit statistics and file rprobs.out contains the fitted retention probabilities.

params.out

```
        Statistics for the two parameter model
        Convergence criteria met in   5 iterations
        Parameter Estimates and Covariances
                              a          =     -0.128805E+02
                              b          =      0.365266E+00
                              var(a)     =      0.403593E+01
                              var(b)     =      0.359176E-02
                              cov(a,b)   =     -0.119846E+00
                              L.25       =      0.322556E+02
                              L.50       =      0.352633E+02
                              L.75       =      0.382710E+02
                              var(L.25)  =      0.311028E+00
                              var(L.50)  =      0.374614E+00
                              var(L.75)  =      0.925268E+00
      Loglikelihood of full model     =     -0.761119E+03
      Loglikelihood of reduced model  =     -0.776399E+03
          Model deviance (  29  df)   =      0.305604E+02
-----------------------------------------------
        Statistics for the three parameter model
        Convergence criteria met in   59 iterations
        Parameter Estimates and Covariances
                              a          =     -0.996696E+01
                              b          =      0.260836E+00
                              p          =      0.584979E+00
                              var(a)     =      0.195978E+01
                              var(b)     =      0.186994E-02
                              var(p)     =      0.888128E-03
                              cov(a,b)   =     -0.597639E-01
                              cov(a,p)   =      0.212890E-01
                              cov(b,p)   =     -0.793638E-03
                              L.25       =      0.339996E+02
                              L.50       =      0.382115E+02
                              L.75       =      0.424234E+02
                              var(L.25)  =      0.844904E+00
                              var(L.50)  =      0.180463E+01
                              var(L.75)  =      0.373951E+01
      Loglikelihood of full model     =     -0.761119E+03
      Loglikelihood of reduced model  =     -0.771826E+03
          Model deviance (  28  df)   =      0.214145E+02
```

rprobs.out

Retention probabilities for 2-parameter model

Length Group	r(1)	Deviance Residual
8.000	0.473277E-04	0.000000E+00
10.000	0.982561E-04	-0.140179E-01
12.000	0.203976E-03	0.000000E+00
14.000	0.423400E-03	-0.290967E-01
16.000	0.878656E-03	-0.725922E-01
18.000	0.182253E-02	-0.120694E+00
20.000	0.377650E-02	-0.212680E+00
22.000	0.780898E-02	-0.278902E+00
24.000	0.160778E-01	0.132641E+01
26.000	0.328127E-01	-0.104772E+01
28.000	0.658016E-01	-0.573864E+00
30.000	0.127582E+00	0.102564E-01
32.000	0.232905E+00	0.109566E+01
34.000	0.386642E+00	-0.116383E+01
36.000	0.566868E+00	-0.369640E+00
38.000	0.730983E+00	0.580434E+00
40.000	0.849431E+00	0.356929E+00
42.000	0.921339E+00	-0.500290E+00
44.000	0.960502E+00	0.872854E+00
46.000	0.980578E+00	0.789487E+00
48.000	0.990550E+00	-0.181160E+00
50.000	0.995426E+00	0.717059E+00
52.000	0.997792E+00	0.116097E+01
54.000	0.998935E+00	0.228979E+01
56.000	0.999487E+00	0.219435E+00
58.000	0.999753E+00	0.130393E+01
60.000	0.999881E+00	0.184331E+01
62.000	0.999943E+00	0.199422E+01
64.000	0.999972E+00	0.153571E+01
66.000	0.999987E+00	0.133090E-04
68.000	0.999994E+00	-0.117741E+01
70.000	0.999997E+00	0.102299E+01
72.000	0.999999E+00	0.000000E+00
74.000	0.999999E+00	0.117741E+01

Retention probabilties for 3-parameter model

Length Group	r(1)	Deviance Residual
8.000	0.377990E-03	0.000000E+00
10.000	0.636690E-03	-0.423562E-01
12.000	0.107226E-02	0.000000E+00
14.000	0.180526E-02	-0.712925E-01
16.000	0.303782E-02	-0.160114E+00
18.000	0.510763E-02	-0.239558E+00
20.000	0.857556E-02	-0.379709E+00
22.000	0.143641E-01	-0.447708E+00
24.000	0.239655E-01	0.830085E+00
26.000	0.397262E-01	-0.136104E+01
28.000	0.651596E-01	-0.982864E+00
30.000	0.105094E+00	-0.309851E+00
32.000	0.165179E+00	0.109690E+01
34.000	0.250018E+00	-0.783019E+00
36.000	0.359659E+00	0.195922E+00
38.000	0.486211E+00	0.936933E+00
40.000	0.614555E+00	0.249043E+00
42.000	0.728727E+00	-0.108014E+01
44.000	0.819039E+00	-0.103803E+00
46.000	0.884067E+00	-0.424265E+00
48.000	0.927788E+00	-0.142195E+01
50.000	0.955844E+00	-0.359575E+00
52.000	0.973313E+00	0.199878E+00
54.000	0.983987E+00	0.157926E+01
56.000	0.990434E+00	-0.543786E+00
58.000	0.994300E+00	0.660717E+00
60.000	0.996609E+00	0.120377E+01
62.000	0.997985E+00	0.140161E+01
64.000	0.998803E+00	0.983981E+00
66.000	0.999289E+00	-0.341705E+00
68.000	0.999578E+00	-0.132603E+01
70.000	0.999749E+00	0.691069E+00
72.000	0.999851E+00	0.000000E+00
74.000	0.999912E+00	0.103558E+01

APPENDIX 4 – PROGRAM TTRAWL.FOR

```
***   This program estimates the parameters of the trouser trawl   ***
***   model proposed by Millar and Walsh (1990). The selection     ***
***   curve is assumed to be the 2 parameter logistic function.    ***
***   Two models can be fitted: the first assumes that the split   ***
***   of fish in the legs of the trawl is 50:50.  The second       ***
***   estimates the split.                                         ***
***   The user may specify that either or both models be fitted.   ***
***                                                                ***
***   The user is prompted for the name of the file containing the ***
***   trouser trawl length frequency data (a maximum of 100 length ***
***   classes).  Each row contains 3 observations -                ***
***   the length class, the number of fish in the wide (test) mesh ***
***   leg, and the number of fish in the fine mesh leg.            ***
***   Computed statistics include the parameters of the fitted     ***
***   logistic selectivity curve, estimates of L.25, L.50 and L.75 ***
***   and their (asymptotic) variances.  These statistics are      ***
***   written to the file params.out.  The fitted selectivity      ***
***   curve (retention probabilities) and the deviance residuals   ***
***   are written to rprobs.out.                                   ***
***                                                                ***
***   The maximization algorithm is Newton-Raphson. The user must  ***
***   specify the maximum number of iterations and the convergence ***
***   tolerance (relative absolute difference between successive   ***
***   parameter estimates), epsilon.  Initial parameter values     ***
***   are derived from user supplied initial guesses of L.25 and   ***
***   L.50                                                         ***
***                                                                ***
***   This program's name is ttrawl.f. It was written by N. Cad-   ***
***   igan, Dept. of Fisheries and Oceans, Government of Canada,   ***
***   St. John's, Newfoundland, Canada, in the fall of 1990. The   ***
***   program is written in standard fortran and was developed on  ***
***   a SUN workstation (UNIX) using a SUN compiler. The program   ***
***   has been successfully compiled and executed on VAX-VMS main- ***
***   frame and on a PC-XT using a Microsoft Fortran compiler.     ***

real*8 n1(100),n2(100),np(100),l(100),l25,l50,xeps
integer nlclass,miter,izero

real*8 param(3),vparam(3,3),lp(3),vlp(3)
real*8 aeps,rll,fll,dll(100),r,c
integer niter,model
       character*50 infile
common /data/ n1,n2,np,xeps,l25,l50,miter,nlclass,l

       call spaces(15)
       write(6,*)'        Input name of data file'
       call spaces(15)
       read(5,*) infile
open(1,file = infile,status='old')
open(2,file = 'params.out')
open(3,file = 'rprobs.out')

***    l(i) is the vector of length classes.
***    n1(i) and n2(i) contain the length frequency data; np(i) is
```

```
***     their sum (i = 1(1)nlclass).
***     nlclass is the number of length classes.
***     izero records the number of groups with np(i) = 0.

izero = 0
nlclass = 1
99 read(1,*,end=100)l(nlclass),n1(nlclass),n2(nlclass)
np(nlclass) = 0.0d0
np(nlclass) = n1(nlclass) + n2(nlclass)
if (np(nlclass).eq.0.0) izero = izero + 1
nlclass = nlclass + 1
        goto 99
100     continue
nlclass = nlclass - 1

call spaces(15)
500 write(6,*)'         Input initial guesses of L.25 and L.50'
call spaces(15)
read(5,*)125,150
if (125 .gt. 150) then
  write(6,*)
  write(6,*)'         Initial guesses are incorrect'
  write(6,*)'         L.25 must be less than L.50'
  write(6,*)'         Re-enter guesses'
  write(6,*)
  goto 500
end if

***     starting values of a and b are computed from 1.25 and 1.50

call spaces(15)
501 write(6,*)'      Input convergence tolerance and maximum'
        write(6,*)'      number of iterations'
call spaces(15)
read(5,*)xeps,miter
if (xeps .le. 0) then
  write(6,*)
  write(6,*)'         Epsilon must be > 0'
  write(6,*)'         Re-enter guesses'
  write(6,*)
  goto 501
end if
if (miter .lt. 0) then
  write(6,*)'         Maximum number of iterations must be >= 0'
  write(6,*)'         Re-enter guesses'
  write(6,*)
  goto 501
end if
call spaces(15)
502 write(6,*)'              Choose which model(s) to fit'
write(6,*)
write(6,*)
write(6,*)'              Enter 1,2 or 3'
write(6,*)
write(6,*)'              1 : 50:50 split'
write(6,*)
write(6,*)'              2 : Estimated split'
write(6,*)
```

```
write(6,*)'                3 : Both 1 and 2'
call spaces(5)
read(5,*)model
if ((model .lt. 1).or.(model .gt. 3)) then
  write(6,*)
  write(6,*)'         Invalid choice'
  write(6,*)'         enter a 1, 2 or 3'
  write(6,*)
  goto 502
end if

if ((model.eq.1).or.(model.eq.3)) then
         goto 101
else
  goto 102
end if

***     This portion fits the 2 parameter model.

101 call twop(param,vparam,niter,aeps,rll,fll,dll)

***     this portion computes the 25%, 50% and 75% retention lengths
***     lp(i) (i = 1(1)3) and their variances vlp(i) (i = 1(1)3).

r = 0.25d0
do 10 i = 1,3
  c = dlog(r/(1.0d0-r))
  vlp(i) = (param(2)**(-4))*(param(2)*param(2)*vparam(1,1) -
     a    (2.0d0*param(2)*(param(1)-c)*vparam(1,2))+
     a    ((param(1)-c)**2)*vparam(2,2))
  lp(i) = (c-param(1))/param(2)
     r = r + 0.25d0
10 continue

write(2,*)
write(2,*)'         Statistics for the two parameter model'
write(2,*)
if (aeps.gt.xeps) then
  write(2,*)'         Maximum number of iterations (',miter,')'
     a             ,'reached'
else
  write(2,*)'         Convergence criteria met in ',niter-1,
     a             ' iterations'
end if
write(2,*)
write(2,*)'         Parameter Estimates and Covariances'
write(2,1002)'a          = ', param(1)
write(2,1002)'b          = ', param(2)
write(2,1002)'var(a)     = ',vparam(1,1)
write(2,1002)'var(b)     = ',vparam(2,2)
write(2,1002)'cov(a,b)   = ',vparam(1,2)
write(2,1002)'L.25       = ',lp(1)
write(2,1002)'L.50       = ',lp(2)
write(2,1002)'L.75       = ',lp(3)
write(2,1002)'var(L.25) = ', vlp(1)
write(2,1002)'var(L.50) = ', vlp(2)
write(2,1002)'var(L.75) = ', vlp(3)
write(2,1002)'Loglikelihood of full model    = ',fll
write(2,1002)'Loglikelihood of reduced model = ',rll
```

```
write(2,*)
write(2,1003)'Model deviance (',nlclass-2-izero,'df) =',
     a  2.0d0*(fll-rll)
write(3,*)
write(3,*)'            Retention probabilities for 2',
     a           '-parameter model'
write(3,*)
write(3,*)'            Length                              Deviance'
write(3,*)'            Group          r(l)                 Residual'
write(3,*)

***     this portion computes the retention probabilities for each
***     length group.

do 11 i = 1,nlclass
  r = dexp(param(1) + param(2)*l(i))/(1.0d0 + dexp(param(1)
     a    + param(2)*l(i)))
  write(3,1000)l(i),r,dll(i)
11 continue

if (model.eq.3) then
  goto 102
else
  goto 103
end if

***     This portion fits the three parameter model

102 call threep(param,vparam,niter,aeps,rll,fll,dll)

***     this portion computes the 25%, 50% and 75% retention lengths
***     lp(i) (i = 1(1)3) and their variances vlp(i) (i = 1(1)3).

r = 0.25d0
do 12 i = 1,3
  c = dlog(r/(1.0d0 - r))
  vlp(i) = (param(2)**(-4))*(param(2)*param(2)*vparam(1,1) -
     a    (2.0d0*param(2)*(param(1)-c)*vparam(1,2))+
     a    ((param(1)-c)**2)*vparam(2,2))
  lp(i) = (c-param(1))/param(2)
     r = r + 0.25d0
12 continue

write(2,*)
write(2,*)
       write(2,*)'-------------------------------------------'
write(2,*)
write(2,*)'         Statistics for the three parameter model'
write(2,*)
if (aeps.gt.xeps) then
  write(2,*)'         Maximum number of iterations (',miter,')'
     a           ,' reached'
else
  write(2,*)'         Convergence criteria met in ',niter-1,
     a           ' iterations'
end if
write(2,*)
write(2,*)'         Parameter Estimates and Covariances'
write(2,1002)'a          = ', param(1)
write(2,1002)'b          = ', param(2)
```

```
write(2,1002)'p          = ', param(3)
write(2,1002)'var(a)     = ',vparam(1,1)
write(2,1002)'var(b)     = ',vparam(2,2)
write(2,1002)'var(p)     = ',vparam(3,3)
write(2,1002)'cov(a,b)   = ',vparam(1,2)
write(2,1002)'cov(a,p)   = ',vparam(1,3)
write(2,1002)'cov(b,p)   = ',vparam(2,3)
write(2,1002)'L.25       = ',lp(1)
write(2,1002)'L.50       = ',lp(2)
write(2,1002)'L.75       = ',lp(3)
write(2,1002)'var(L.25) = ', vlp(1)
write(2,1002)'var(L.50) = ', vlp(2)
write(2,1002)'var(L.75) = ', vlp(3)
write(2,1002)'Loglikelihood of full model    = ',fll
write(2,1002)'Loglikelihood of reduced model = ',rll
write(2,*)
write(2,1003)'Model deviance (',nlclass-3-izero,'df) =',
     a  2.0d0*(fll-rll)
write(3,*)
write(3,*)'             Retention probabilties for 3',
     a           '-parameter model'
write(3,*)
write(3,*)'          Length                                  Deviance'
write(3,*)'          Group              r(l)                 Residual'
write(3,*)

***     this portion computes the retention probabilities for each
***     length group.

do 13 i = 1,nlclass
  r = dexp(param(1) + param(2)*l(i))/(1.0d0 +
     a       dexp(param(1) + param(2)*l(i)))
  write(3,1000)l(i),r,dll(i)
13 continue

        write(6,*)
        write(6,*)' Program finished:'
        write(6,*)'  - estimated statistics are in file params.out'
        write(6,*)'  - estimated selectivity curve(s) and deviance ',
     a            'residuals'
        write(6,*)'    are in file rprobs.out'

1000    format(11x,f7.3,5x,e14.6,5x,e14.6)
1001    format(/,a37)
1002    format(/,a40,3x,e14.6)
1003    format(a28,1x,i3,1x,a6,4x,e14.6)
103 stop
end

***--------------------------------------------------------------------

subroutine twop(p,v,iter,eps,rl,fl,dll)

***     This routine computes the estimates of a and b (p - fixed at
***      0.5) for the two parameter model. Parameter estimates are
***     returned invector p, their variances in v. The loglikelihood
***     of the fitted model is returned in rl; that of the full model
***     in fl. The number of iterations to convergence and the last
```

```
***     computed epsilon are returned in iter and eps respectively.
***     Note: dl a 2x1 vector of 1st order partial derivatives of the
***       likelihood function.(regardless of dimension).
***           xj a 2x2 matrix of 2nd order partial derivatives of the
***           likelihood function.(regardless of dimension).
***           t's are temporary variables.
***           nd is the dimension of the problem i.e. 2.

real*8 n1(100),n2(100),np(100),l(100),l25,l50,xeps
integer nlclass,miter

real*8 p(3),dl(3),xj(3,3),v(3,3)
real*8 eps,rl,fl,t1,t2,t3,t4,t5,det,dll(100),trl,tfl
integer iter,nd
common /data/ n1,n2,np,xeps,l25,l50,miter,nlclass,l

nd = 2
eps = 1.0d0
iter = 0
p(2) = -1.0986123d0/(l25-l50)
p(1) = -p(2)*l50

10 if ((eps.gt.xeps).and.(iter.le.miter)) then

    do 11 i = 1,3
    dl(i)=0.0d0
    xj(1,i)=0.0d0
    xj(2,i)=0.0d0
    xj(3,i)=0.0d0
11   continue

  do 12 i = 1,nlclass
    t1 = 1.0/(2.0d0 + dexp(-p(1)-p(2)*l(i)))
    t2 = 1.0d0 - t1
    t3 = 1.0d0 - 2.0d0*t1
    t4 = t1*t3/(t2**2)
    t5 = 2.0d0*(t1**2) - 4.0d0*t1 + 1.0d0

    dl(1) = dl(1) + t3*(n1(i) - n2(i)*t1/t2)

    dl(2) = dl(2) + l(i)*t3*(n1(i) - n2(i)*t1/t2)

    xj(1,1) = xj(1,1) - t4*(n1(i) + np(i)*t5)

    xj(2,2) = xj(2,2) - l(i)*l(i)*t4*(n1(i) + np(i)*t5)

    xj(1,2) = xj(1,2) - l(i)*t4*(n1(i) + np(i)*t5)

12   continue
  xj(2,1) = xj(1,2)

  call mxint(xj,nd,3,det)
  if(det.eq.0.0d0) then
            write(6,*)'solution diverges, try different'
     write(6,*)'starting values of L.25 and L.50'
     stop
  end if
  p(1) = p(1) - xj(1,1)*dl(1) - xj(1,2)*dl(2)
  p(2) = p(2) - xj(2,1)*dl(1) - xj(2,2)*dl(2)

  eps = dabs(max((xj(1,1)*dl(1) + xj(1,2)*dl(2))/p(1),
     a     (xj(2,1)*dl(1) + xj(2,2)*dl(2))/p(2)))

  iter = iter + 1
```

```
goto 10
endif

rl = 0.0d0
fl = 0.0d0
do 14 i = 1,3
  do 13 j = 1,3
    v(i,j) = 0.0d0
13   continue
14 continue

do 15 i = 1,nlclass
    dll(i) = 0
    t1 = 1.0d0/(2.0d0 + dexp(-p(1)-p(2)*l(i)))
    rl = rl + n1(i)*dlog(t1) + n2(i)*dlog(1.0d0-t1)
    trl = n1(i)*dlog(t1) + n2(i)*dlog(1.0d0-t1)
    tfl = 0
    t2 = 0
    if (np(i).gt.0.0) t2 = n1(i)/np(i)
    if ((n1(i).gt.0.0).and.(n2(i).gt.0.0)) then
      t2 = n1(i)/np(i)
      fl = fl + n1(i)*dlog(t2) + n2(i)*dlog(1.0d0-t2)
      tfl = n1(i)*dlog(t2) + n2(i)*dlog(1.0d0-t2)
    end if
    dll(i) = sign(1,t2-t1)*dsqrt(2.0d0*(tfl-trl))

  t3 = np(i)*dexp(p(1)+p(2)*l(i))/((1.0d0+dexp(p(1)
     a    +p(2)*l(i)))*((1.0d0 + 2.0d0*dexp(p(1)+p(2)*l(i)))**2))
  v(1,1) = v(1,1) + t3
  v(1,2) = v(1,2) + l(i)*t3
  v(2,2) = v(2,2) + l(i)*l(i)*t3
15 continue
v(2,1) = v(1,2)

call mxint(v,nd,3,det)

return
end

*** ----------------------------------------------------------------------
subroutine threep(p,v,iter,eps,rl,fl,dll)

***     This routine computes the estimates of a, b and p (the
***     proportion of fish entering the wide mesh).
***     Parameter estimates are returned in vector p, their
***     variances in v. The loglikelihood of the estimated model
***     is returned in rl; that of the full model in fl. The number of
***     iterations to convergence and the last computed epsilon are
***     returned in iter and eps respectively.

***     Note: dl a 3x1 vector of 1st order partial derivatives of the
***       likelihood function.
***           xj a 3x3 matrix of 2nd order partial derivatives of the
***           likelihood function.
***           t's are temporary variables.
***           nd is the dimension of the problem i.e. 3.

real*8 n1(100),n2(100),np(100),l(100),125,150,xeps
integer nlclass,miter

real*8 p(3),dl(3),xj(3,3),v(3,3)
real*8 eps,rl,fl,t1,t2,t3,t4,det,dll(100),trl,tfl
```

```
      integer iter,nd
      common /data/ n1,n2,np,xeps,l25,l50,miter,nlclass,l

      nd = 3
      eps = 1.0d0
      iter = 0
      p(2) = -1.0986123d0/(l25-l50)
      p(1) = -p(2)*l50
      p(3) = 0.5

   10 if ((eps.gt.xeps).and.(iter.le.miter)) then

          do 11 i = 1,3
          dl(i)=0.0d0
          xj(1,i)=0.0d0
          xj(2,i)=0.0d0
          xj(3,i)=0.0d0
   11     continue

        do 12 i = 1,nlclass
          t1 = p(3)/(1.0d0 + (1.0d0 - p(3))*dexp(-p(1)-p(2)*l(i)))
          t2 = 1.0d0 - t1
          t3 = p(3) - t1

          t4 = t3/p(3)
          dl(1) = dl(1) + t4*((n2(i)*t1/t2) - n1(i))
          dl(2) = dl(2) + l(i)*t4*((n2(i)*t1/t2) - n1(i))

          t4 = p(3)*(1.0d0 - p(3))
          dl(3) = dl(3) + t4*((t1*np(i)) - n1(i))

          t4 = t1*t3/(t2*p(3)*p(3))
          xj(1,1) = xj(1,1) + t4*(n1(i) - np(i)*t1 + (n2(i)*t3/t2))
          xj(2,2) = xj(2,2) + l(i)*l(i)*t4*(n1(i) - np(i)*t1 +
           a                (n2(i)*t3/t2))
          xj(1,2) = xj(1,2) + l(i)*t4*(n1(i) - np(i)*t1 +
           a                (n2(i)*t3/t2))

          t4 = p(3)*p(3)/((1.0d0-p(3))**2)
          xj(3,3) = xj(3,3) + t4*((2.0d0*p(3)*(np(i)*t1 - n1(i)))
           a      + n1(i) - (np(i)*t1*t1))

          t4 = p(3)*p(3)/(1.0d0-p(3))
          xj(1,3) = xj(1,3) + t4*t1*t3*np(i)
          xj(2,3) = xj(2,3) + l(i)*t4*t1*t3*np(i)
   12     continue
        xj(2,1) = xj(1,2)
        xj(3,1) = xj(1,3)
        xj(3,2) = xj(2,3)

        call mxint(xj,nd,nd,det)
        if(det.eq.0.0d0) then
                  write(6,*)'solution diverges, try different'
          write(6,*)'starting values of L.25 and L.50'
          stop
        end if

        p(1) = p(1) - xj(1,1)*dl(1) - xj(1,2)*dl(2) - xj(1,3)*dl(3)
        p(2) = p(2) - xj(2,1)*dl(1) - xj(2,2)*dl(2) - xj(2,3)*dl(3)
        p(3) = p(3) - xj(3,1)*dl(1) - xj(3,2)*dl(2) - xj(3,3)*dl(3)
```

```
  eps = dabs(max((xj(1,1)*dl(1)+xj(1,2)*dl(2)+
     a   xj(1,3)*dl(3))/p(1),
     a    (xj(2,1)*dl(1)+xj(2,2)*dl(2)+xj(2,3)*dl(3))/p(2),
     a   (xj(3,1)*dl(1)+xj(3,2)*dl(2)+xj(3,3)*dl(3))/p(3)))

  iter = iter + 1
goto 10
endif

rl = 0.0d0
fl = 0.0d0
do 14 i = 1,3
  do 13 j = 1,3
    v(i,j) = 0.0d0
13   continue
14 continue

do 15 i = 1,nlclass
  dll(i) = 0
  t1 = p(3)*dexp(p(1)+p(2)*l(i))/(1.0d0 - p(3) + dexp(p(1)
     a          + p(2)*l(i)))
  t2 = p(3)*p(3)*(1.0d0-t1)
  t3 = p(3)*p(3)*(1.0d0-p(3))

    rl = rl + n1(i)*dlog(t1) + n2(i)*dlog(1.0d0-t1)
    trl = n1(i)*dlog(t1) + n2(i)*dlog(1.0d0-t1)
    tfl = 0
    t4 = 0
    if (np(i).gt.0.0) t4 = n1(i)/np(i)
    if ((n1(i).gt.0.0).and.(n2(i).gt.0.0)) then
      t4 = n1(i)/np(i)
      fl = fl + n1(i)*dlog(t4) + n2(i)*dlog(1.0d0-t4)
      tfl = n1(i)*dlog(t4) + n2(i)*dlog(1.0d0-t4)
    end if
    dll(i) = sign(1,t4-t1)*dsqrt(2.0d0*(tfl-trl))

  v(1,1) = v(1,1) + np(i)*((p(3) - t1)**2)*t1/t2
  v(2,2) = v(2,2) + l(i)*l(i)*np(i)*((p(3) - t1)**2)*t1/t2
  v(2,1) = v(2,1) + l(i)*np(i)*((p(3)-t1)*(p(3)-t1))*t1/t2

  v(3,3) = v(3,3) + np(i)*(1.0d0 - t1)*t1/(t3*(1.0d0 - p(3)))

  v(3,1) = v(3,1) + np(i)*(p(3) - t1)*t1/t3
  v(3,2) = v(3,2) + l(i)*np(i)*(p(3) - t1)*t1/t3
15 continue

v(1,2) = v(2,1)
v(1,3) = v(3,1)
v(2,3) = v(3,2)

call mxint(v,nd,nd,det)

return
end

*** ------------------------------------------------------------------

subroutine  mxint(a,nrr,nra,det)
        dimension isw(96), a(*)
        double precision a,det

***     this matrix inversion subroutine takes a  nra*nra
***     square matrix and inverts the upper  nrr*nrr
```

```
***     submatrix

        na = nra+1
        nd = nrr*nra
        do 1 i=1,nrr
    1   isw(i) = i
        ndiag = 1
        nup = 1
        det = 1.
        do 9 i=1,nrr
           ka = i
           am = 0.
           la = ndiag
           do 2 j=i,nrr
              if (dabs(a(la)).le.am) go to 2
              ka = j
              am = dabs(a(la))
    2      la = la+1
           if (am.gt.1.0e-20) go to 3
  write(6,*)'pivot equal zero'
           det = 0.
           return
    3      if (ka.eq.i) go to 5
           mup = i
           do 4 mlow=ka,nd,nra
              s = a(mup)
              a(mup) = a(mlow)
              a(mlow) = s
    4      mup = mup+nra
           is = isw(ka)
           isw(ka) = isw(i)
           isw(i) = is
    5      aa = a(ndiag)
           det = det*aa
           do 6 j=i,nd,nra
    6      a(j) = a(j)/aa
           a(ndiag) = 1./aa
           nel = nup
           do 8 j=1,nrr
              if (j.eq.i) go to 8
              aa = a(nel)
              a(nel) = 0.
              ka = i
              do 7 k=j,nd,nra
                 a(k) = a(k)-aa*a(ka)
    7         ka = ka+nra
    8      nel = nel+1
           nup = nup+nra
    9   ndiag = ndiag+na
        i = 1
   10   if (isw(i).ne.i) go to 11
        i = i+1
        if (i.gt.nrr) go to 13
   11   j = isw(i)
        isw(i) = isw(j)
        isw(j) = j
        mlow = (i-1)*nra+1
        mup = (j-1)*nra+1
        do 12 k=1,nrr
           s = a(mlow)
```

```
            a(mlow) = a(mup)
            a(mup) = s
            mlow = mlow+1
   12     mup = mup+1
          go to 10

   13     return
          end

***  ----------------------------------------------------------------------
subroutine spaces(k)
integer k
  do 10 i = 1,k
    print*,' '
10   continue
return
end
```